

REMARKS

This is a full and timely response to the outstanding non-final Office Action mailed April 30, 2004. Reconsideration and allowance of the application and pending claims are respectfully requested.

Abstract Objection

The abstract of the disclosure has been objected to because the current abstract does not reflect the inventive feature of the claimed invention to distinguish over the prior art. Applicant respectfully traverses.

The abstract of the disclosure currently provides:

The present disclosure relates to a system and method for dynamically patching code. In one arrangement, the system and method pertain to intercepting program instructions, determining if a program instruction requires unavailable hardware functionality, and dynamically replacing the program instruction with a replacement instruction that does not require unavailable hardware functionality if it is determined that the program instruction requires unavailable hardware functionality.

From this excerpt, it is apparent that Applicant is describing a system and method that is described in Applicant's specification. Therefore, the abstract of the disclosure is supported by the specification. In terms of distinguishing over the prior art, Applicant respectfully submits that the inventive embodiment described in the abstract of the disclosure does distinguish over the prior art. For this reason, Applicant has not amended the abstract of the disclosure. If required by the Examiner, Applicant will

amend the abstract of the disclosure to reflect limitations of an allowed claim, once such allowability has been identified.

Claim Rejections - 35 U.S.C. § 102(b)

Claims 1-22 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Dockser (U.S. Pat. No. 5,862,370). In addition, claims 1-22 have been separately rejected under 35 U.S.C. § 102(b) as being anticipated by Giles, et al. ("Giles," U.S. Pub. No. 2002/0133810). Applicant respectfully traverses both rejections.

As is apparent from the foregoing, each remaining independent claim has been amended. Applicant therefore respectfully submits that the above-noted rejections are moot as having been directed to Applicant's claim in their original form. However, Applicant discusses the Dockser and Giles references and their applicability to Applicant's claims in the following for the Examiner's consideration.

It is axiomatic that "[a]nticipation requires the disclosure in a single prior art reference of *each element* of the claim under consideration." W. L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 1554, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983)(emphasis added). Therefore, every claimed feature of the claimed invention must be represented in the applied reference to constitute a proper rejection under 35 U.S.C. § 102(b).

Referring first to claim 13, not every feature of claim 13 is disclosed by Dockser or Giles. In claim 13, Applicant recites the following (emphasis added):

13. A dynamic patching program stored on a computer-readable medium, the program comprising:

logic configured to gain control over execution of a program;

logic configured to intercept original program instructions during program execution;

logic configured to determine if associated instructions have been cached in a code cache controlled by the dynamic patching program;

logic configured to execute associated instructions that have been cached in the code cache;

logic configured to determine if an original program instruction requires unavailable hardware functionality; and

logic configured to dynamically replace original program instructions with replacement instructions that do not require unavailable hardware functionality if it is determined that the program instructions require unavailable hardware functionality, wherein the logic configured to dynamically replace is configured to fetch a replacement instruction, ***store the replacement instruction in the code cache,*** and ***execute the replacement instruction from the code cache,*** wherein the logic configured to dynamically replace does not replace or translate original program instructions that do not require unavailable hardware functionality.

Contrary to that claimed by Applicant, the Dockser system does not include “logic configured to determine if associated instructions have been cached in a code cache controlled by the dynamic patching program” or “logic configured to execute associated instructions that have been cached in the code cache”. Instead, the Dockser system uses a filter that substitutes a call instruction so that a substitution routine is called by the computer hardware instead of the original routine. Dockser, column 5, lines 32-40. Because the Dockser system operates in this manner, Dockser also does

not teach or suggest logic configured to “store the replacement instruction in the code cache” or “execute the replacement instruction from the code cache”.

Regarding the Giles reference, Giles does not teach a system that “does not replace or translate original program instructions that do not require unavailable hardware functionality”, as is required by claim 13. To the contrary, the Giles system translates each instruction block. As described by Giles (Giles, paragraph 0033, lines 1-5):

The dynamic translation step 112 occurs even if the instruction block includes no instructions that are potentially disruptive, i.e., the instruction block includes only benign instructions that are unlikely to affect the operation or integrity of the host computer system.

Such a manner of operation could potentially decrease execution performance by creating additional processing.

Referring now to Applicant’s claim 17, recited is (emphasis added):

17. A method for dynamically patching code, comprising:
gaining control over the execution of a program using a software interface;

intercepting original program instructions during execution of the program using the software interface;

determining whether associated instructions have been cached in a code cache of the software interface and, if so, ***executing the cached instructions from the code cache***;

if associated instructions have not been cached, determining if the original program instructions require unavailable hardware functionality; and

dynamically replacing the original program instructions with replacement instructions that do not require unavailable hardware functionality if it is determined that the original program instructions require unavailable hardware functionality, the dynamic replacing comprising fetching replacing instructions, *storing the replacement instructions in the code cache*, and *executing the replacement instructions from the code cache*;

wherein original program instructions that do not require unavailable hardware functionality are not replaced with replacement instructions or translated.

For reasons discussed in the foregoing, Dockser fails to teach or suggest any of “determining whether associated instructions have been cached in a code cache of the software interface”, “executing the cached instructions from the code cache”, “storing the replacement instructions in the code cache”, or “executing the replacement instructions from the code cache”.

For reasons discussed in the foregoing, Giles fails to teach or suggest a method wherein “original program instructions that do not require unavailable hardware functionality are not replaced with replacement instructions or translated”.

Lastly, referring to Applicant’s claim 21, recited is (emphasis added):

21. *A dynamic execution layer interface (DELI) residing between an application and computing system hardware*, comprising:

a transparent mode layer that is configured to gain control over the operation of the application and to *fetch replacement instructions* that are to replace existing application instructions;

a system control and configuration layer configured to provide policies for the replacement of existing application instructions with the replacement instructions;

a *core* configured to, during execution of the application, receive policies for the replacement of existing application instructions, to determine whether the existing application instructions require unavailable hardware functionality, to *receive replacement instructions to be executed in lieu of the existing application instructions* that do require unavailable hardware functionality, *to cache the replacement instructions*, and to *execute the replacement instructions, wherein existing application instructions that do not require unavailable hardware functionality are not translated or replaced*; and

a *code cache* in which the replacement instructions are cached and from which the replacement instructions are executed.

Referring first to Dockser, Applicant notes that Dockser fails to teach or suggest a “dynamic execution layer interface (DELI) residing between an application and computing system hardware” that comprises each of a “transparent mode layer”, a “system control and configuration layer”, a “core”, and a “code cache”. In addition, Dockser does not teach or suggest an interface component that is configured to “fetch replacement instructions”, to “receive replacement instructions to be executed in lieu of the existing application instructions”, to “to cache the replacement instructions”, or to “execute the replacement instructions”.

As for the Giles reference, Giles similarly does not teach or suggest a “dynamic execution layer interface (DELI) residing between an application and computing system hardware” that comprises each of a “transparent mode layer”, a “system control and configuration layer”, a “core”, and a “code cache”. Moreover, Giles does not teach or suggest an interface wherein “existing application instructions that do not require unavailable hardware functionality are not translated or replaced”.

Due to the above-noted shortcomings of the Dockser and Giles references, Applicant respectfully asserts that neither Dockser nor Giles anticipates Applicant's claims. Therefore, Applicant respectfully requests that the rejection of the claims under both references be withdrawn.

Canceled Claims

As identified above, claims 1-7, 9-12, 14-15, and 18 have been canceled from the application through this response without prejudice, waiver, or disclaimer. Applicant reserves the right to present these canceled claims, or variants thereof, in continuing applications to be filed subsequently.

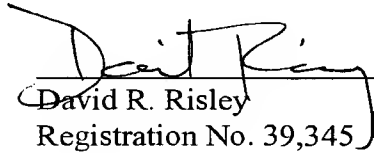
New Claims

As identified above, claims 23-27 have been added into the application through this response. Applicant respectfully submits that these new claims describe an invention novel and unobvious in view of the prior art of record and, therefore, respectfully requests that these claims be held to be allowable.

CONCLUSION

Applicant respectfully submits that Applicant's pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned attorney at (770) 933-9500.

Respectfully submitted,


David R. Risley
Registration No. 39,345

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postage prepaid, in an envelope addressed to: Assistant Commissioner for Patents, Alexandria, Virginia, 22313-1450, on

July 14, 2004
Mary Megan
Signature